

INCREMENT and DECREMENT

RISK ASSESSMENT: *Low*

SAFETY
CONSIDERATIONS:
Electrical Shock

~~Hazard~~
ENVIRONMENTAL
Electrostatic
CONSIDERATIONS:
Discharge
None

TERMINAL LEARNING OBJECTIVE

ACTION: Explain how the 8086 microprocessor performs arithmetic operations.

CONDITIONS: Given a classroom environment, Computer Aided Instruction (CAI), workstation, NIDA electronic trainer, and official OMEMS reference

STANDARDS: Correctly answer 18 of 25 questions on the 8086 microprocessor arithmetic operations.

ENABLING LEARNING OBJECTIVE

ACTION: Explain how the 8086 microprocessor performs increment and decrement operations.

CONDITIONS: Given a classroom environment, Computer Aided Instruction (CAI), workstation, NIDA electronic trainer, and official OMEMS reference

STANDARDS: Correctly answer 18 of 25 questions on the 8086 microprocessor arithmetic operations.

INCREMENT

- ◆ SPECIAL ARITHMETIC INSTRUCTION.
- ◆ ADDS A 1 TO THE CONTENTS OF A REGISTER OR MEMORY LOCATION.
- ◆ USED IN COUNTERS.

DECREMENT

- ◆ SPECIAL ARITHMETIC INSTRUCTION.
- ◆ SUBTRACTS A 1 FROM THE CONTENTS OF A REGISTER OR MEMORY LOCATION.
- ◆ USED IN COUNTERS.

FLAGS

- ◆ THE INCREMENT AND DECREMENT INSTRUCTION **DOES NOT AFFECT THE CARRY FLAG.**

INCREMENT

INCREMENT BX REGISTER

BX = FFFF FFFF

+ 1

BX = 0000

BX = 0000 0000

+ 1

BX = 0001

DECREMENT

DECREMENT BX REGISTER

$$\begin{array}{r} \text{BX} = 0000 \qquad 0000 \\ \qquad \qquad \qquad - \quad 1 \\ \hline \text{BX} = \text{FFFF} \end{array}$$

$$\begin{array}{r} \text{BX} = \text{FFFF} \qquad \text{FFFF} \\ \qquad \qquad \qquad - \quad 1 \\ \hline \text{BX} = \text{FFFE} \end{array}$$

INC reg (Increment Register)

CODE

MNEMONIC

PSEUDOCODE

43

INC BX

INCREMENT
REGISTER BX

INC M (Increment Memory Directly)

(byte 3)(byte 4) $\leftarrow$ (byte 3)(byte 4) + 1

CODE

MNEMONIC

PSEUDOCODE

FF

INC M

INCREMENT

06

MEMORY

00

LOCATION

10

1000

CF351K05-11

INC M (Increment Memory Indirectly)

OPCODE PSEUDOCODE

MNEMONIC

C7
immediate into

MVI M, 1234

Move

06
location

memory

50

5050, 1234

50

34

12

BE
immediate into

MVI, SI 5050

Move

50

the SI, 5050

50

CF351K05-12

DEC reg (Decrement Register)

CODE
PSEUDOCODE

MNEMONIC

4B

DEC BX

DECREMENT
REGISTER BX

DEC M (Decrement Memory Directly)

(byte 3)(byte 4) $\longleftarrow$ (byte 3)(byte 4) - 1

CODE

MNEMONIC

PSEUDOCODE

FF

DEC M

DECREMENT

0E

MEMORY

00

LOCATION

10

1000

CF351K05-14

DEC M (Decrement Memory Indirectly)

OPCODE PSEUDOCODE

MNEMONIC

C7
immediate into

MVI M, 1234

Move

06
location

memory

50

5050, 1234

50

34

12

BE
immediate into

MVI, SI 5050

Move

50

the SI, 5050

50

CF351K05-15

FF

DEC M (5050)

INCREMENT AND DECREMENT OPERATION

Starting with address *3000h*, load *SI* with *5050h*.

Indirectly, transfer the contents of memory location *5050h* into *AX*. **Increment** the *AX*.

Indirectly, increment the contents of memory location *5050h*. **Increment** the contents of *SI* **twice**.

Indirectly, transfer the contents of memory location *5052h* into the *BX*. **Decrement** the contents of the *BX*. **Directly decrement** the contents of memory location *5052h*. Halt the program.

<u>ADDRESS</u>	<u>OPCODE</u>	<u>MNEMONIC</u>
3000		MVI SI, 5050H
3001		
3002		
3003		MOV AX, M(5050h)
"Indirect"		
3004		
3005		INC AX
3006		INC M(5050h) "Indirect"
3007		
3008		INC SI
3009		INC SI
300A		MOV BX, M (5052h)
"Indirect"		
300B		
300C		DEC BX
300D		DEC M (5052h)
300E		
300F		
3010		
3011		HALT
3012		

ENABLING LEARNING OBJECTIVE

ACTION: Explain how the 8086 microprocessor performs addition and subtraction operations.

CONDITIONS: Given a classroom environment, Computer Aided Instruction (CAI), workstation, NIDA electronic trainer, and official OMEMS reference

STANDARDS: Correctly answer 18 of 25 questions on the 8086 microprocessor arithmetic operations.

ADDITION

In Addition you can add.....

- ◆ BYTE TO BYTE
- ◆ WORD TO WORD
- ◆ BYTE TO WORD (FILL HB W/ZEROS)

ADDITION

In Addition you can add.....

- ◆ MEMORY TO REGISTER
- ◆ REGISTER TO MEMORY
- ◆ REGISTER TO REGISTER
- ◆ IMMEDIATE NUMBER TO MEM OR REG.

ADDITION

- ◆ CARRY FLAG
 - WILL SET IF A CARRY FROM MOST SIGNIFICANT BIT OCCURS DURING AN ADDITION.
 - WILL SET OR RESET AT THE END OF EACH INSTRUCTION
- ◆ ZERO FLAG
 - WILL SET IF THE RESULT OF AN ADDITION IS ZERO
 - WILL SET OR RESET AT THE END OF EACH INSTRUCTION

ADDITION

♦ DECIMAL SYSTEM

- ADMISSIBLE MARKS ARE 0-9
- HIGHEST ADMISSIBLE MARK IS 9
- A **CARRY** IS GENERATED IF WE GO ABOVE 9

EXAMPLE:

$$\begin{array}{r} 9 \\ +1 \\ \hline \end{array}$$

Carry  10

ADDITION

◆ HEXADECIMAL SYSTEM

- ADMISSIBLE MARKS ARE 0-F
- HIGHEST ADMISSIBLE MARK IS F
- A **CARRY** IS GENERATED IF WE GO ABOVE F

EXAMPLE:

$$\begin{array}{r} F \\ + 1 \\ \hline \end{array}$$

Carry  10

DOUBLE PRECISION ADDITION

- ◆ DIVIDE TWO 32 BIT WORDS INTO FOUR 16 BIT WORDS.
- ◆ STORE THE 16 BIT WORDS INTO 4 REGISTERS
- ◆ **ADD** THE LOWER WORDS, **ADC** THE HIGHER WORDS.
- ◆ STORE THE ANSWER IN 2 REGISTERS.

DOUBLE PRECISION ADDITION

HIGH WORDS

XXXX

“ADC”

XXXX

XXXX

LOW WORDS

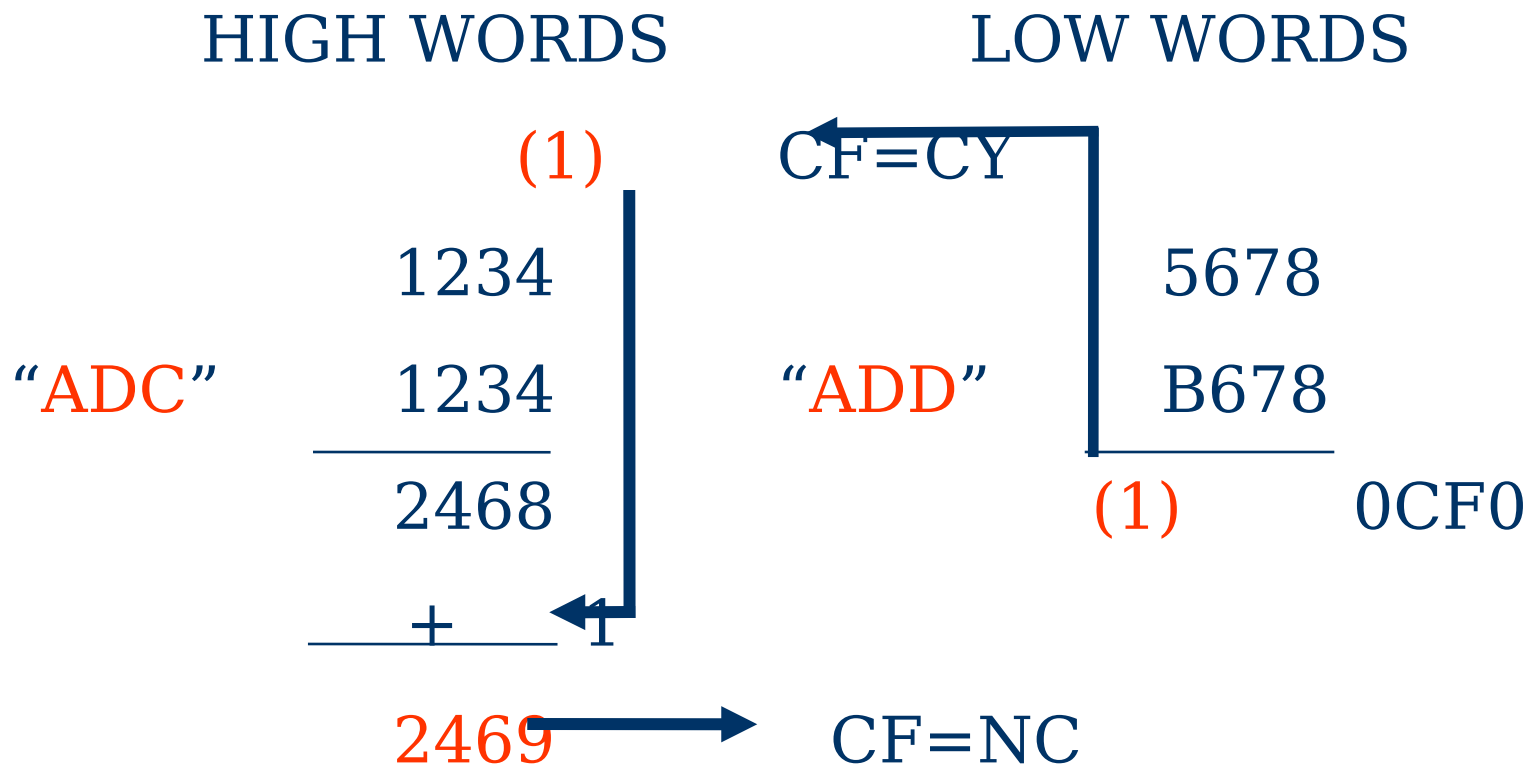
XXXX

“ADD”

XXXX

XXXX

DOUBLE PRECISION ADDITION



ADD reg (Add Register to Register)

CODE
PSEUDOCODE

MNEMONIC

03	ADD AX, BX	Add register
C3		to AX

ADC reg (Add Register to Register with Carry)

CODE
PSEUDOCODE

MNEMONIC

13
BX C3
Carry

ADC AX, BX

Add register
to AX plus
Flag

ADDITION OPERATION

- ◆ Starting at address *1100h*, load *AX* with **DC88h**. Load *BX* with **1A45h**. Load *CX* with **24EFh**. Add contents of *CX* to contents of *AX*. Add contents of *BX* to contents of *AX* with carry. Halt the program.

ADDRESS

OPCODE

MNEMONIC

1100

MVI AX, DC88h

1101

1102

1103

MVI BX, 1A45h

1104

1105

1106

MVI CX, 24EFh

1107

1108

1109

ADD AX, CX

110A

110B

ADC AX, BX

110C

110D

HALT

110E

SUBTRACTION

In subtraction you can subtract.....

- ◆ BYTE FROM BYTE
- ◆ WORD FROM WORD
- ◆ BYTE FROM WORD (FILL HB W/ZEROS)

SUBTRACTION

In subtraction you can subtract.....

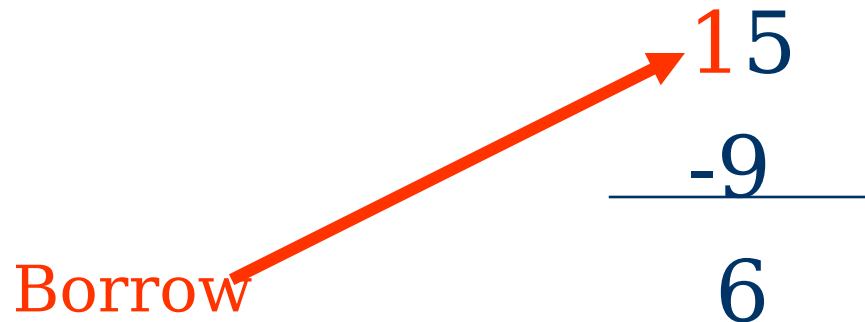
- ◆ MEMORY FROM REGISTER
- ◆ REGISTER FROM MEMORY
- ◆ REGISTER FROM REGISTER
- ◆ IMMEDIATE NUMBER FROM MEM OR REG

SUBTRACTION

- ◆ CARRY FLAG
 - WILL SET IF A BORROW FROM MOST SIGNIFICANT BIT OCCURS DURING A SUBTRACTION.
 - WILL SET OR RESET AT THE END OF EACH INSTRUCTION
- ◆ ZERO FLAG
 - WILL SET IF THE RESULT OF A SUBTRACTION IS ZERO
 - WILL SET OR RESET AT THE END OF EACH INSTRUCTION

SUBTRACTION

- ◆ IN THE DECIMAL SYSTEM
A **BORROW** OF 10 OCCURS
EXAMPLE:



A subtraction problem is shown with a horizontal line. Above the line is the number 15, and below the line is the number -9. Below the line is the number 6. A red arrow points from the word "Borrow" to the '1' in 15.

$$\begin{array}{r} 15 \\ -9 \\ \hline 6 \end{array}$$

Borrow

SUBTRACTION

- ◆ IN THE HEXADECIMAL SYSTEM A BORROW OF 16 OCCURS.
IF A **BORROW** FROM THE MSB OCCURS THE CARRY FLAG SETS.
EXAMPLE:

1 **BORROW (SETS CARRY)**

A

- F _____

B

DOUBLE PRECISION SUBTRACTION

- ◆ DIVIDE TWO 32 BIT WORDS INTO FOUR 16 BIT WORDS.
- ◆ STORE THE 16 BIT WORDS INTO 4 REGISTERS
- ◆ SUB LOWER WORDS, SBB HIGHER WORDS
- ◆ STORE ANSWER IN 2 REGISTERS.

DOUBLE PRECISION SUBTRACTION

HIGH WORDS

XXXX

“SBB”

XXXX

XXXX

XXXX

LOW WORDS

XXXX

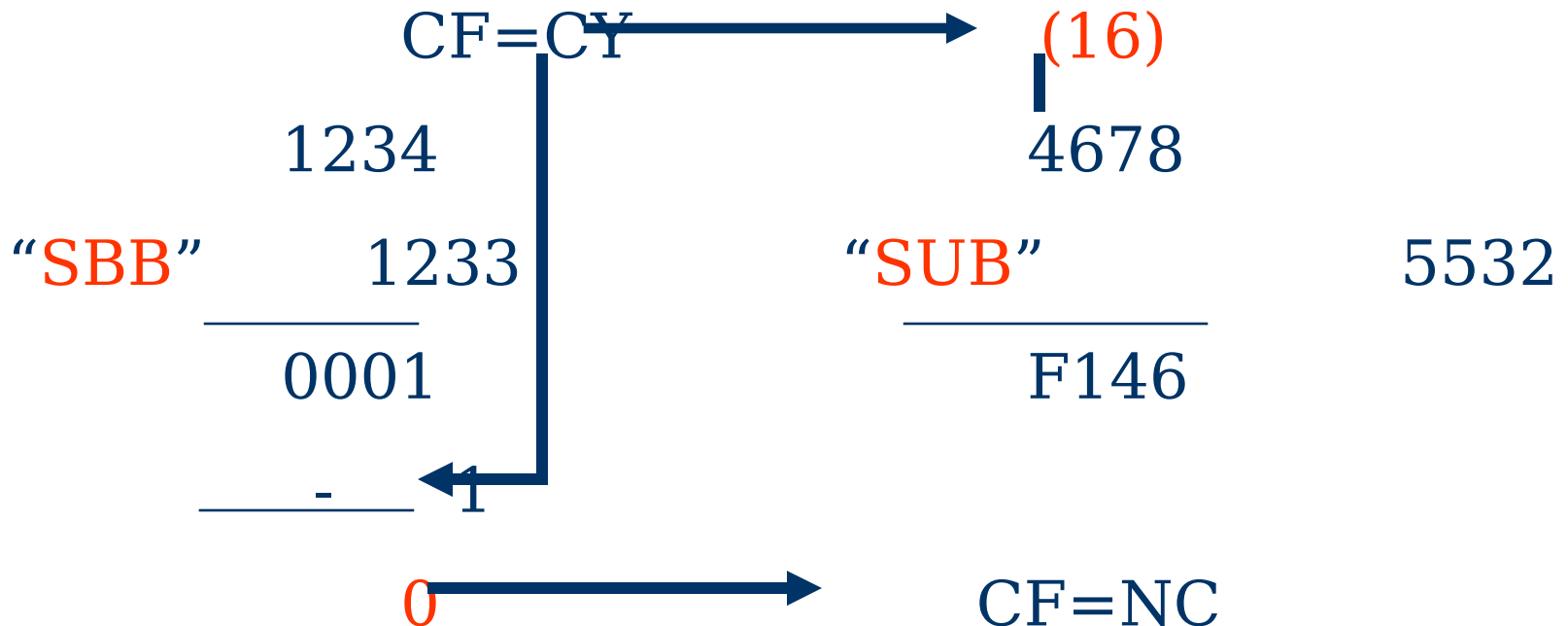
“SUB”

XXXX

DOUBLE PRECISION SUBTRACTION

HIGH WORDS

LOW WORDS



SUB reg (Subtract Register from Register)

CODE
PSEUDOCODE

MNEMONIC

2B
register
AX
C3

SUB AX, BX

Subtract
BX from

SUB reg (Subtract Register from Register with Borrow)

CODE
PSEUDOCODE

MNEMONIC

1B
register C3
Carry
from
register AX

SBB AX, BX

Subtract
BX and
Flag

SUBTRACTION OPERATION

Starting with address *2200h*, load *AX* with **D126h**. Load *BX* with **E114h**. Load *CX* with **F011h**. Subtract contents of *CX* from contents of *AX*. Subtract contents of *BX* from contents of *AX* with borrow. Halt the program.

ADDRESS

OPCODE

MNEMONIC

2200

MVI AX, D126h

2201

2202

2203

MVI BX, E114h

2204

2205

2206

MVI CX, F011h

2207

2208

2209

SUB AX, CX

220A

220B

SBB AX, BX

220C

220D

HALT

220E

Question

- ♦ What is the second step required when we add 32 bit numbers with the 8086?
- ♦ A. Load the 16 bit numbers into 4 registers
- ♦ B. Divide the 32 bit numbers into 16 bit numbers.
- ♦ C. ADD the lower words, ADC the higher words

Answer

- ♦ What is the second step required when we add 32 bit numbers with the 8086?
- ♦ *A. Load the 16 bit numbers into 4 registers*
- ♦ B. Divide the 32 bit numbers into 16 bit numbers.
- ♦ C. ADD the lower words, ADC the higher words

Question

- ♦ Why is it necessary to use a SBB instruction when subtracting the higher words?
- ♦ A. To account for any carries that occurred
- ♦ B. To account for any borrows that occurred.
- ♦ C. To confuse the microprocessor.

ANSWER

- ♦ Why is it necessary to use a SBB instruction when subtracting the higher words?
- ♦ A. To account for any carries that occurred
- ♦ *B. To account for any borrows that occurred.*
- ♦ C. To confuse the microprocessor.

ENABLING LEARNING OBJECTIVE

ACTION: Program Computer
Increment/Decrement,
and Subtraction Operations.

Addition

CONDITIONS: Given a classroom
environment,

Computer

**Aided Instruction (CAI),
workstation, NIDA electronic trainer,
and official OMEMS reference**

STANDARDS: Correctly answer 18 of 25
questions on the **8086**
microprocessor arithmetic operations.

QUESTION

- ♦ What flag is not affected by the increment instruction?
- ♦ A. The Zero Flag
- ♦ B. The Trap Flag
- ♦ C. The Carry Flag
- ♦ D. The Sign Flag

ANSWER

- ◆ What flag is not affected by the increment instruction?
- ◆ A. The Zero Flag
- ◆ B. The Trap Flag
- ◆ *C. The Carry Flag*
- ◆ D. The Sign Flag

QUESTION

- ◆ What is double-precision addition used for?
- ◆ A. To verify the correct results of an addition.
- ◆ B. To add numbers twice as fast.
- ◆ C. To add numbers that are larger than our ALU

ANSWER

- ◆ What is double-precision addition used for?
- ◆ A. To verify the correct results of an addition.
- ◆ B. To add numbers twice as fast.
- ◆ *C. To add numbers that are larger than our ALU*

SUMMARY

- ♦ **INCREMENT INSTRUCTIONS**
- ♦ **DECREMENT INSTRUCTIONS**
- ♦ **ADDITION INSTRUCTIONS**
- ♦ **SUBTRACTION INSTRUCTIONS**